A.0005 数组与字符串

笔者终于又坐下来继续写这个 Perl Guide 了. 今天我们来讲讲数组[array]和字符串.

学过中等数学的同学应该知道,当我们需要表示一长串类似的变量的时候,我们会用到"下标",比如有个数列,我们会用 a0,a1,a2,a3,…,an 来表示.我们使用下标的原因在于,需要记录的变量比较多,无暇给他们依次命名(比如 a,b,c,…,j,k…).有了下标,也方便我们来计算数列的长度和随机查找.

在 Perl 编程里面,也可以定义类似的带下标的数组.下面我们就来看看如何创建一个数组,如何访问数组,如何对数组进行操作维护.(注:参考某网友的总结)

创建数组:

```
@a=(1,2,3,4,5,6,7);
@b=(1..7);
@c=("a".."z");
@d=();
$num=@b;
```

上面使用了四种方法分别构造了四个数组.第一种,一看就意会了吧,就是建一个数组:@a,一共有 7 个元素,分别是 1 到 7.第二种,效果和第一种相同,非常适用于创建一个很大的数组,比如 1~10000 的数组.第三种,和前两种类似,不过是换成了字符.创建之后,数组@c 的第一个元素就是 a,第二个是 b,以此类推,最后一个是 z.第四种,就是创建一个空的数组@d,之后呢可以通过添加操作慢慢往里面添加想要的元素.

第五行,这个内容没处加了就加在这里吧.这个操作可以获取数组@b 的元素个数(长度).$num 里放的数就是@b 的长度,这里$num 就等于 7,因为@b 里面有七个元素,没啥好多说的.

访问数组:

```
$b[0]=10000;
@b[5,7,9]=(45,56,67);
```

注意看第一行,这就代表你想用数组@b 的第一个元素的时候,写$b[0]就好了.Perl 中的数组下标是从 0 开始的.一个 N 个元素的数组@b,元素分别是$b[0],$b[1]直到$b[N-1].注意这里引导符变成了$不是@。通过这个变化表示这时正在访问一个单独的数据。而不是一组数据。

第二行,这是使用多个下标，但是这时就要使用@引导符了。代表着给@b 的下标为 5,7,9 的元素赋值.下标都是写在方括号里面.

数组的操作:

```
@d=@c;
@c=(@c,53);
push(@c,53);
push(@c,(45,64));
push(@c,@a);
```

第一行,整体复制,不难理解,就是照着@c 复制一个一样的@d.

第二行,叫做追加,在原来@c 的后面加一个新元素 53.看看也很形象,c 等于(c,53).

第三行,和第二行是等效的.push 是一个专用的追加操作函数,在末尾压入一个新元素.

第四行是告诉我们,可以在一个数组后面接上另一个数组(接上一截).这个就是在@c 后面接上了(45,64)这个数组,使得@c 多了两个元素.

第五行,照着前面的意会,就是把@a 接在了@c 后面.如果领会了含义应该知道,这时候@c 后面多了一段@a,而@a 自身并没有变化.

下面还有一些操作,或许会用到,我就大段抄别人码的字了,供参考吧:

pop 就是从 Perl 数组的最后取出一个元素。用法为:pop(@a);

这个取出,有两个含义.@a 中的最后一个元素被删除了;我们可以用一个变量来接着这个被删去的变量:$b=pop(@a);

unshift 函数可以从 Perl 数组的开头加入元素用法为:

```
unshift(@c,"hello");
unshift(@c,("hello","halloha"));
unshift(@c,@a);
```

shift 的功能是从 Perl 数组的开头取出一个元素。用法为：shift(@a);

有了操作 Perl 数组两端的函数那么一定也会有操作 Perl 数组中间部分的函数，这个函数就是 splice.splice 函数有三个作用。第一个作用是向 Perl 数组中间的一部分插入内容。例如:

```
@d=(1..9);
my@e=("a".."f");
splice(@d,2,2,@e);
```

将会得到 12abcdef56789，注意这里是从第二个开始插入，不是从下标为 2 的元素开始插入的。

splice 的第二个功能是删除，例如在刚才的代码上面再加上:

```
splice(@d,2,6);
```

将可以得到 1256789.

splice 的第三个功能就是删除到末尾。语法为:

```
splice(@d,2);
```

就是从第二个开始，删除到末尾。

现在讲讲字符串.其实笔者比较讨厌字符串,也不擅长,对于字符串操作反应总是慢别人半拍到一拍.

字符串就是一串字符,什么字符都行,一般用一个变量存放.比如,$str1=”nihao, 2huo! hahaha!”.字符串记着用双引号引起来,以此来告诉电脑,这是一整个变量,是个字符串.

现在讲讲字符串的操作.

♦字符串合并.用句号,就是点.例如:

```
$s=”wole”.”gequ”;
```

这样子$s 就变成了”wolegequ”.点也可以用在变量之间或者常量变量之间,例如:

```
$s=$s1.$s2;
```

♦取字符串的长度.length(字符串名)例如:

```
$str="abCD99e";
$strlen=length($str);
```

这样$strlen 就等于 7.

♦取字符串的子串(或单个字符).substr(串名,位置,长度),位置从下标 0 开始.例如:

```
$str="ABCDEFG1234567";
$a=substr($str,0,5);
```

这样, $a 等于 ABCDE.同理,当第三个参数,长度,为 1 时,就是单独取一个字符出来.

★字符串分割. @数组=split（pattern,串）将 Perl 字符串按照某种分隔符分成多个字串.这些字串赋给左边的数组,数组中一个元素就是一个字串.举个例子:

```
$str="ABCDE FG12  34567";
@array=split(/\s+/,$str);
```

这样子$array[0]就等于”ABCDE”, $array[1]等于”FG12”, $array[1]等于”34567”.理解了吧.

如何来表示想以什么分隔符来分割呢,这个比较麻烦,大体说来,就是在//之间写上分隔符.用什么分隔符都行,比如你可以把字母 a 当成分隔符.笔者只会用上述例子里那个,\s+,意义是一个或几个空格或 tab 或回车.这个挺常用的,基本可以通用了=.=,会用这个就差不多了其实. 如果自己用的分隔符很奇怪,不知道怎么写,可以上网查. 如果想学写,可以参考这个链接:http://t.cn/8FOgQsn 或者翻书什么的.